Implementasi dan Analisis Simulasi QOS dan Perfomance Device dengan Menggunakan ONOS dan Iperf3

I Putu Agus Eka Pratama[1], **I Made Adhiarta Wikantyasa**[2]

[1,2]Teknologi Informasi, Fakultas Teknik Universitas Udayana, Jl. Raya Kampus Unud, Bukit Jimbaran, Badung, Bali, Indonesia
e-mail: [1]eka.pratama@unud.ac.id, [2]madhiarta@gmail.com

Abstract

Internet have evolved so rapidly in this day and age from where user only can access to internet service through cable UTP(LAN) and now user can access it from every where through nirkable (Wireless Network) and from this improvement it caused lots of device connected and heavy traffic coming in and out from a network. Because of this situation the network industry come with a new idea using a new network architecture that known as Software Defined Network (SDN). Software Defined Network (SDN) is a new way to manage, design and implementing network architecture where data flow from the control plane separated from the hardware. Network industry believe that SDN can change the network architecture that people using now that known as Traditonal Network because SDN can overcome the weakness of Tradional network that tend to be closed and distributed can be changed by SDN to be open source, can be programmed and can be controlled centrally. In this research will be implementing and analys simulation Quality of Service (QOS) and Perfomance device openflow switch using ONOS as controller to monitor the device perfomance and iperf3 as a Quality Of Service testing.

Keywords: *Wireless, Software Defined Network, Open Network Operating System (ONOS), Quality of Service (QOS),* Iperf3

1. Pendahuluan

Pada saat ini Internet sudah berkembang sangat pesat dikarenakan 2 hal yaitu berkembangnya Teknologi jaringan dan teknologi informasi yang sangat menunjak. Dapat dikatakan seperti itu dikarenakan saat ini internet telah diangap sebagai sumber daya yang sangat penting bagi manusia untuk aktivitas sehari-harinya, karena dengan adanya internet ini seorang pengguna dapat mengakses informasi dan aktivitas lainnya cukup melalui internet tanpa melakukannya secara manual selain dari itu pengguna dapat juga membagi resource pada pengguna lain yang ada pada jaringan internet ini. Dengan adanya perkembangan ini akan menyebabkan jumlah perangkat terhubung dan jumlah trafik pada jaringan meningkat sangat pesat yang akan susah dikelolah pada aristektur jaringan konvensional ini, maka dari itu industri jaringan memiliki ide untuk menggunakan arsitektur jaringan yang baru yaitu *Sofware Defined Network* (SDN).

Software Defined Network (SDN) merupakan jaringan arsitektur yang bekerja dibawah kendali software sebagai kontrol utama. SDN memerlukan beberapa metode agar kontrol plane untuk berkomunikasi dengan data plane. SDN memungkinkan network administrator untuk memprogram pusat kontrol jaringan melalui sebuah controller tanpa akses fisik ke switch. Dengan adanya SDN ini dapat menggantikan arsitektur jaringan yang saat ini dipakai atau yang biasa disebut *Traditional Network* / jaringan konvensional dikarenakan sifatnya Jaringan konvensional yang cenderung tertetutup serta terdistribusi akan dapat diubah oleh SDN untuk menjadi terbuka (*Open Source*), karena dapat diprogram dan dikontrol secara terpusat.

Software Defined Network (SDN) di control secara terpusat dengan *controller* yang berfungsi sebagai pengontrol alur dari tiap packet yang dikirim dan ditereima pada jaringan SDN ini. *Controller* yang digunakan adalah *Open Network Operating System* (ONOS) *controller* ini memberikan *high-availability* (ketersediaan yang tinggi), *scalable*, dan *perfomance* sekelas carrier *provider* (AT&T, NTT *Communication*). Dengan semua penjelasan tersebut penulis membuat penelitian yang berjudul "Implementasi Dan Analisis Simulasi QOS dan *Perfomance Device* Dengan Menggunakan ONOS dan Iperf3" dikarenakan ONOS saja blum cukup untuk melakukan test perfomance pada jaringan ditambahkan tools Iperf3 untuk membantunya dalam test *Quality of Service* (QOS) pada jaringan.

2. Metode Penelitian

Metode penelitian merupakan metode yang akan digunakan saat dilakukannya penelitian ini yaitu dengan menggunakan metode eksperimen dimana dilakukannya percobaan dan analisis terhadap QOS dan Perfomance device openflow switch dengan menggunakan ONOS dan iperf3 ini

2.1 Tahap Penelitian.

Tahap Penelitian pada penelitian ini dapat dilihat pada flowchart berikut.

Gambar 1 Flowchart Tahap Pengujian

Gambar 1 merupakan alur pengujian ini yang dimulai dari instalasi ONOS dan iperf 3 yang akan dijadikan sebagai controller dan Quality Of Service testing. Tahap berikutnya melakukan konfigurasi Path dari ONOS dan iperf 3 yang berfungsi untuk menjalankannya pada sistem operasi Ubuntu ini dan untuk tahap berikutnya melakukan pengujian Qualiy Of Service(QOS) menggunakan iperf3 dan tahap terahir melakukan Analisis perfomance device Openflow Switch pada jaringan SDN yang dibuat dengan menggunakan ONOS.

2.2 Analisa Kebutuhan Perangkat

Analisa Kebutuhan Perangkat merupakan perangkat yang akan digunakan selama penelitian ini. Berikut merupakan perangkat keras dan perangkat lunak yang digunakan.

2.2.1 Perangkat Keras

Penggunaan perangkat keras yang digunakan hanya menggunakan laptop dengan spesifikasi sebagai berikut.

1. Lenovo ideapad
2. *Processor*: Intel(R) Core (TM) i5-6200U CPU @ 2.30GHz (4CPUs)
3. Memory DDR4 8000MB RAM
4. Storage HDD 1TB

2.2.2 Perangkat Lunak

Perangkat lunak yang digunakan pada penilitan antara lain.

1. Sistem operasi menggunakan linux Ubuntu 18.04.2 LTS
2. JDK versi 8
3. Mininet
4. ONOS(Open Network Operating System)

3. Perancangan dan Pembahasan

Perancangan dan pembahasa terkait dengan instalasi dari aplikasi ONOS dan iperf serta pembuatan topology yang digunakan pada mininet serta pengujian qos dan analisis perfomance device dengan menggunakan ONOS dan Iperf3.

3.1 Instalasi

Instalasi yang dilakukan pada perangkat lunak JDK, mininet dan onos. Berikut tahapan yang dilakukan untuk melakukan instalasi tersebut.

3.1.2 Instalasi JDK 8

Instalasi JDK versi 8 dilakukan pada sistem operasi Ubuntu 18.04.2 LTS dikarenakan lisensi oracle yang berubah maka download instalasi JDK akan dilakukan melalui website seperti berikut.

Java SE Development Kit 8u211

You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.

◉ Accept License Agreement ○ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.86 MB	jdk-8u211-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.76 MB	jdk-8u211-linux-arm64-vfp-hflt.tar.gz
Linux x86	174.11 MB	jdk-8u211-linux-i586.rpm
Linux x86	188.92 MB	jdk-8u211-linux-i586.tar.gz
Linux x64	171.13 MB	jdk-8u211-linux-x64.rpm
Linux x64	185.96 MB	jdk-8u211-linux-x64.tar.gz
Mac OS X x64	252.23 MB	jdk-8u211-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	132.98 MB	jdk-8u211-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.18 MB	jdk-8u211-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.57 MB	jdk-8u211-solaris-x64.tar.Z
Solaris x64	91.93 MB	jdk-8u211-solaris-x64.tar.gz
Windows x86	202.62 MB	jdk-8u211-windows-i586.exe
Windows x64	215.29 MB	jdk-8u211-windows-x64.exe

Gambar 2 Website Oracle

Gambar 2 merupakan gambaran website oracle yang dimana user harus terima lisensi untuk melanjutkan download file tersebut. Tahap berikutnya akan melakukan ekstrak file JDK dengan sintaks berikut ini.

```
tar xzvf jdk-8u211-linux-x64.tar.gz
```

Kode Program 1. Sintaks file bashrc

Kode Program 1 merupakan sintaks terminal untuk ekstrak file yang telah di download tersebut. Untuk tahap berikutnya akan dilakukan konfigurasi PATH java dengan membukak file berikut ini.

```
nano. Bashrc
```

Kode Program 1 Sintaks file bashrc

Kode Program 2 merupakan sintaks terminal untuk membuka *file* bashrc untuk melakukan konfigurasi JDK dengan sistem operasi ubuntu. Tahapan berikutnya akan melakukan konfigurasi PATH JAVA_HOME pada *file* bashrc.

```
Export
JAVA_HOME=/home/Jackson/Desktop/jdk
-8u211-linux-x64/jdk1.8.0_211

export
PATH=$JAVA_HOME/bin:$PATH
```

Kode Program 2 Konfigurasi OpenJDK

Kode program 3 merupakan sintaks terminal untuk melakukan konfigurasi PATH JAVA_HOME sehingga dapat menjalankan aplikasi ONOS.

3.1.3 Instalasi Mininet

Instalasi mininet dilakukan pada sistem operasi Ubuntu 18.04.2 LTS dengan menggunakan sintaks instalasi pada terminal seperti berikut ini.

```
sudo apt-get install mininet
```

Kode Program 3. Sintaks Mininet

Kode Program 4 merupakan sintaks terminal untuk menginstall mininet dengan automatis pada sistem operasi Ubuntu ini.

3.1.4 Instalasi *Open Network Operating System*(ONOS)

Instalasi *Open Network Operating System* (ONOS) pada versi falcon 1.5.2 dilakukan pada sistem operasi ubuntu dengan menggunakan sintaks terminal seperti berikut.

```
    wget
http://archive.apache.org/dist/kara
f/3.0.5/apache-karaf-3.0.5.tar.gz
    wget
http://archive.apache.org/dist/mave
n/maven-3/3.3.9/binaries/apache-
maven-3.3.9-bin.tar.gz
```

Kode Program 4. Sintaks Unduh ONOS

Kode Program 5 merupakan sintaks untuk unduh *Open Network Operating System* dan menempatkannya pada file *Downloads*. Tahap berikutnya akan melakukan ekstrak *file* yang telah di *download* seperti berikut ini.

```
    tar -zxvf apache-karaf-
3.0.5.tar.gz -C ../Applications/
    tar -zxvf apache-maven-3.3.9-
bin.tar.gz -C ../Applications/
```

Kode Program 5. Sintaks Ekstrak ONOS

Kode Program 6 bertujuan untuk melakukan ekstrak ONOS dan menempatkannya pada file Applications agar mudah untuk diakses. Tahapan berikutnya akan melakukan penambahan fitur onos pada file org.apache.karaf.features.cfg yang terdapat pada *directory* Applications/apachekaraf3.0.5/etc tambahan fitur ini akan di tempatkan pada bagian akhir dari featuresRepositories pada line 39 berikut merupakan script fitur yang akan ditambahkan.

```
      mvn:org.onosproject/onos-
        features/1.5.0-
      SNAPSHOT/xml/features
```

Kode Program 6. Sintaks fitur onos

Kode Program 7 merupakan sintaks yang digunakan untuk menambahkan repository fitur onos pada karaf. Untuk tahap berikutnya akan melakukan konfigurasi onos dengan membukak file berikut ini.

```
nano. Bashrc
```

***Kode Program 7**. Sintaks Menjalankan ONOS*

Kode Program 8 merupakan sintaks terminal untuk membuka *file* bashrc untuk melakukan konfigurasi ONOS pada sistem operasi ubuntu. Tahapan berikutnya akan menambahkan PATH Maven dan karaf serta menentukan nama user dan IP onos yang dapat dilihat seperti berikut ini

```
export ONOS_ROOT=~/onos
source
$ONOS_ROOT/tools/dev/bash_profile
KARAF_ROOT=/root/Applications/apach
e-karaf-3.0.5
```

```
exportM2_HOME=/root/Applications/ap
ache-maven-3.3.9
export ONOS_USER=root
export ONOS_GROUP=root
export ONOS_IP=10.0.0.30
```

Kode Program 8. Sintaks ONOS

Kode Program 9 merupakan sintaks yang ditambahkan pada file bashrc ini yang menunjukkan PATH lokasi dari maven dan karaf serta juga konfigurasi nama user da ip onos. Pada tahap selanjunya akan melakukan installasi onos dengan command berikut ini.

```
mvn clean install
```

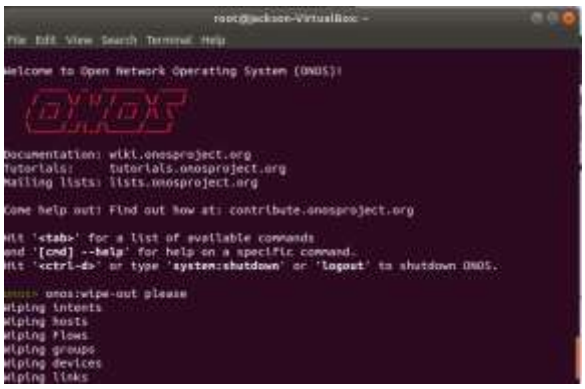
Kode Program 10. Sintaks Installasi

Kode Program 10 merupakan sintasks yang digunakan untuk melakukan instalasi onos pada operating sistem ubuntu ini Untuk tahap selanjutnya akan menjalankan onos dengan menggunakan sintaks berikut ini.

```
ok clean
```

Kode Program 11. Sintaks Memulai ONOS

Kode Program 11 ini merupakan sintaks yang digunakan untuk memulai ONOS jika semua sudah berjalan dengan lancar onos akan tertampil seperti gambar berikut ini.



Gambar 3 ONOS running

Gambar 3 menunjukkan bahwa onos telah berjalan dengan lancar pada sistem operasi Ubuntu 18.04.2 LTS ini.

3.1.5 Instalasi Iperf3

Instalasi Iperf pada versi 3 dilakukan pada sistem operasi ubuntu dengan menggunakan sintaks terminal seperti berikut.

```
sudo apt-get install iperf3
```

Kode Program 12. Sintaks Instalasi Iperf3

Kode Program 12 merupakan fungsi untuk melakukan instlasi Iperf yang berguna untuk mengukur kualitas network pada *software defined network* mininet.

3.2 Pengujian

Setelah selesainnya instalasi dari seluruh perangkat lunak yang digunakan akan dilakukan pengujian. Pengujian ini akan dibagi menjadi 2 yaitu pengujian pada QOS (quality of service) menggunakan iperf3 dan untuk keduanya melakukan analisis perfomance dari device openflow switch pada jaringan SDN dengan menggunakan onos.

3.3 Pengujian QOS menggunakan iperf3.

Pada pengujian *Quality Of Service* (QOS) pada pengujian ini akan dilakukan perfomance testing dari *custom* topology yang dibuat menggunakan Bahasa python dengan mininet API. Berikut merupakan script python yang digunakan.

```
from mininet.node import
CPULimitedHost
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.log import
setLogLevel, info
from mininet.node import
RemoteController
from mininet.cli import CLI
from mininet.link import TCLink

class MinimalTopo( Topo ):
"Minimal topology with a single
switch and two hosts"

            def build(self):
          # Create two hosts.
       h1  = self.addHost( 'h1' )
       h2  = self.addHost( 'h2' )
       h3  = self.addHost( 'h3' )
       h4  = self.addHost( 'h4' )
       h5  = self.addHost( 'h5' )
       h6  = self.addHost( 'h6' )
       h7  = self.addHost( 'h7' )
       h8  = self.addHost( 'h8' )
       h9  = self.addHost( 'h9' )
      h10 = self.addHost( 'h10' )
      h11 = self.addHost( 'h11' )
      h12 = self.addHost( 'h12' )
      h13 = self.addHost( 'h13' )
      h14 = self.addHost( 'h14' )
      h15 = self.addHost( 'h15' )
      h16 = self.addHost( 'h16' )
      h17 = self.addHost( 'h17' )
      h18 = self.addHost( 'h18' )

          # Create a switch
```

```
        s1 = self.addSwitch( 's1' )
        s2 = self.addSwitch( 's2' )
        s3 = self.addSwitch( 's3' )
        s4 = self.addSwitch( 's4' )
        s5 = self.addSwitch( 's5' )
        s6 = self.addSwitch( 's6' )
        s7 = self.addSwitch( 's7' )
        s8 = self.addSwitch( 's8' )
        s9 = self.addSwitch( 's9' )
       s10 = self.addSwitch( 's10' )

       self.addLink( s2, s1 )
      self.addLink( s3, s1 )
      self.addLink( s4, s1 )
      self.addLink( s5, s2 )
      self.addLink( s6, s2 )
      self.addLink( s7, s3 )
      self.addLink( s8, s3 )
      self.addLink( s9, s4 )
      self.addLink( s10, s4 )
      self.addLink( h1, s5 )
      self.addLink( h2, s5 )
      self.addLink( h3, s5 )
      self.addLink( h4, s6 )
      self.addLink( h5, s6 )
      self.addLink( h6, s6 )
      self.addLink( h7, s7 )
      self.addLink( h8, s7 )
      self.addLink( h9, s7 )
      self.addLink( h10, s8 )
      self.addLink( h11, s8 )
      self.addLink( h12, s8 )
      self.addLink( h13, s9 )
      self.addLink( h14, s9 )
      self.addLink( h15, s9 )
      self.addLink( h16, s10 )
      self.addLink( h17, s10 )
      self.addLink( h18, s10 )

      topos = {
      'minimal': MinimalTopo
                  }
```

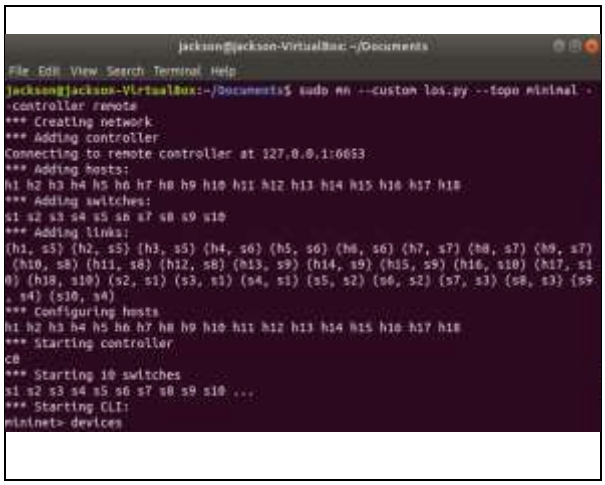
Kode Program 13. *Script topo.py*

Kode program 13 merupakan script python yang berfungsi untuk membuat topology yang akan digunakan dengan menggunakan API mininet dari custom topology itu akan dibuat sebuah jaringan SDN yang berisi 18 host dam 10 switch yang dimana mereka telah di setting untuk berhubung satu sama lain. Tahap berikutnya adalah untuk menjalankan script tersebut menggunakan sintaks seperti berikut ini.

```
Sudo mn –custom topo.py –topo
minimal –controller remote
```

Kode Program 14. Script topo.py

Kode program 14 merupakan sintaks untuk mulai build topology custom mininet dengan menggunakan script python yang telah dibuat. Setelah menjalankan sintaks itu pada terminal akan menampilkan gambar seperti berikut ini.



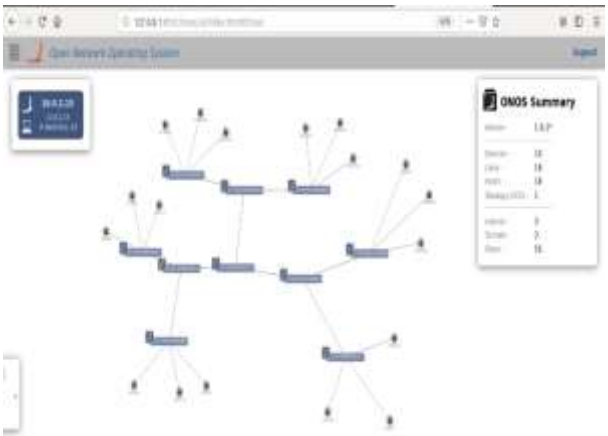
Gambar 3 *Ouput* Sintaks Mininet

Gambar 3 menunjukkan hasil output dari sintaks yang telah dijalankan yang menunjukkan mininet telah berhasil membuat jaringan SDN yang berisi host 18 dan switch 10. Tahap berikutnya akan menampilkan hosts yang telah dibuat oleh mininet pada ONOS dengan menggunakan sintaks seperti berikut ini.

```
pingall
```

Kode Program 15. Sintaks *Pingall User*

Kode program 15 menunjukkan sintaks yang digunakan untuk ping semua host yang ada untuk supaya dapat ditampilkan pada web gui ONOS. Setelah selesai hosts akan terlihat seperti gambar berikut ini.



Gambar 5 Web GUI ONOS

Gambar 5 menunjukkan hasil dari sintaks yang dijalankan tersebut pada gambar ini terlihat bahwa topology telah berhasil ditampilkan pada onos yang dapat dilihat pada kanan ONOS summary terlihat bahwa topology ini berisi 18 host yang telah terhubung dengan 18 flows dan pada kiri atas juga dapat dilihat ip controller onos dan switch 10. Setelah itu akan mulai testing QOS pada topology ini dengan menjalankan sintaks berikut ini.

```
Iperf3 -c 10.0.0.2 -t 20 -p 5566
dan
iperf3 -s -I 1 -p 5566
```

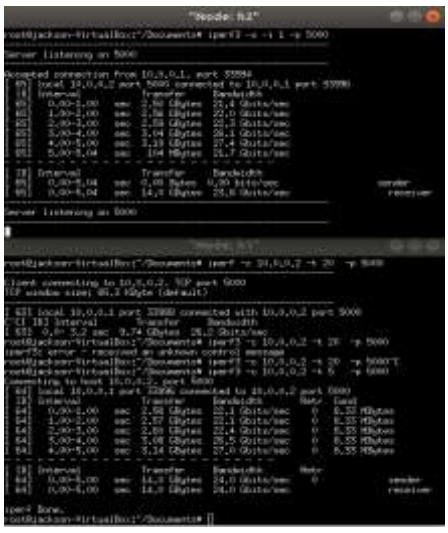
Kode Program 15. Sintaks Iperf3 TCP

Kode Program 15 merupakan sintaks yang digunakan pada iperf3 untuk mengirimkan packet TCP pada host 1 ke host 2 dan untuk menjalankan host menjadi *server mode*. Berikut merupakan sintaks untuk mengirmkan packet UDP pada iperf3.

```
Iperf3 -c 10.0.0.2 -t 20 -p 5577
-u -b 10M
dan
  iperf3 -s -I 1 -p 5577
```

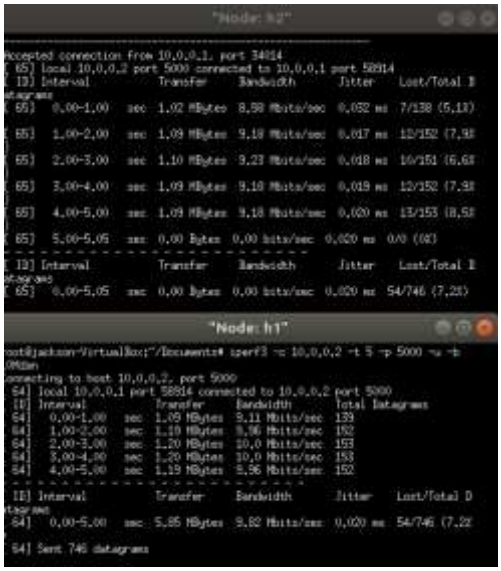
Kode Program 16. Sintaks Iperf3 TCP

Kode Program 16 merupakan sintaks yang digunakan pada iperf3 untuk mengirimkan packet UDP pada host 1 ke host 2 dan untuk menjalankan host menjadi server mode. Berikut merupakan output dari sintaks" berikut.



Gambar 5 Ouput Sintaks Iperf3 TCP

Gambar 5 menunjukkan hasil output dari sintaks yang telah dijalankan yang menunjukkan bahwa host 1 bertindak sebagai client yang mengirimkan packet TCP yang berjumlah 2 packet pada host 2 yang bertindak sebagai server. Berikut merupakan output dari sintaks UDP.



Gambar 6 output sintaks UDP

Gambar 6 menunjukkan hasil output dari sintaks yang telah dijalankan yang menunjukkan bahwa host 1 bertindak sebagai *client* yang mengirimkan packet UDP yang berjumlah 2 *packet* pada *host* 2 yang bertindak sebagai *server*. Berikut merupakan *output* dari sintaks UDP.

3.2.2 Analisis Perfomance Device menggunakan ONOS.

Pada pengujian ini akan menganalisi perfomance terhadap packet" yang dikirim tersebut pada devices Openflow switch yang ada pada topology pada aplikasi ONOS. Fitur ini dapat diakses melalui REST API Onos yang dapat dilihat link seperti berikut.

```
http://127.0.0.1:8181/onos/v1/docs
```

Kode program 17 Sintaks activate app

Kode program 17 merupakan links yang digunakan untuk mengakses REST API pada ONOS. Setelah itu pada REST API ini pilih opsi seperti gambar ini.

Gambar 6 REST API ONOS

Gambar 6 menunjukkan opsi yang harus dipilih untuk melihat perfomance terhadap device openflow pada jaringan SDN. Setelah itu akan disuruh untuk menginputkan "ID DEVICE" seperti gambar berikut ini.

Gambar 7 Input device id

Gambar 7 menunjukkan form pada opsi statistic/port/{*device id*} yang dimana user harus menginputkan *device id* dahulu sebelum melanjutkannya. *Device id* tersebut dapat dilihat menggunakan sintaks berikut ini.

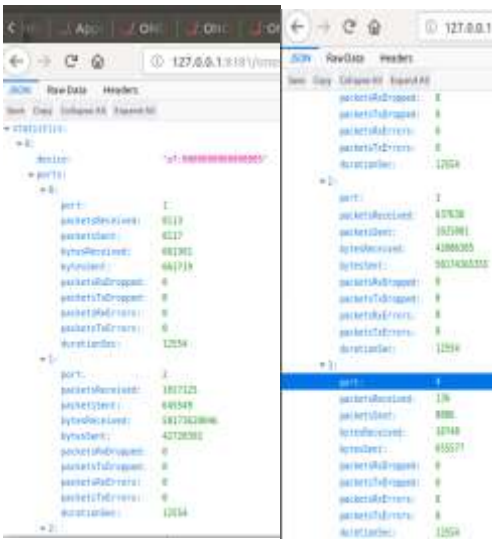
```
Devices
```

Kode program 18 Sintaks devices

Kode program 18 merupakan sintaks yang digunakan untuk menampilkan seluruh devices openflow switch yang ada pada jaringan SDN ini. Jika sudah dijalankan akan muncul seperti gambar berikut ini.

Gambar 8 output sintaks devices

Gambar 8 menunjukan hasil dijalankannya sintaks devices yang menampilkan seluruh devices openflow switch yang ada pada jaringan SDN ini. Selanjutnya akan melihat perfomance device yang dipilih pada REST API ini seperti berikut.



Gambar 9 Rest API statistic device

Gambar 9 menunjukan hasil statistic yang diperoleh oleh device openflow switch 5 pada tiap port yang merupakan hasil total packet" data yang dikirim saat melakukan pengujian pertama.

4. Hasil dan Pembahasan

Table 1 Quality Of Service(QOS)

Host	*Jitter*	*Packet loss*	UDP	UTP
1	0.190	7.5	2,3819	1.403
2	0.173	7.7	2.3819	6.681
3	0.039	7.6	2.3819	6.582
4	0.028	7.6	2.3819	10.657
5	0.208	7.3	2.3819	11.538
6	0.40	6.3	2.3819	5.131
7	0.154	7.4	2.3819	12.294
8	0.022	7.2	2.3819	15.257
9	0.029	7.2	2.3819	14.867
10	0.257	7.7	2.3819	13.956
11	0.256	7	2.3819	14.379
12	0.040	7.5	2.3819	14.050
13	0.282	7.5	2.3819	15.367
14	0.020	7.6	2.3819	14.961
15	0.279	7.7	2.3819	13.926
16	0.222	7.7	2.3819	14.096
17	0.021	7.5	2.3819	15.418
18	0.233	7.7	2.3819	13.861
Rata-Rata	0.1585	7.428	2.3819	11.912

Hasil yang didapatkan memalui kedua pengujian ini adalah pada pengujian pertama meperoleh nilai packet loss, jitter, bandwith, *throughput* UTP dan UDP yang dimana sebagai parameter dari analisis *quality of service* jaringan SDN yang diperoleh dengan menggunakan iperf3 dan yang kedua memperoleh *total packet* yang dikirim pada pengujian pertama pada tiap port *device* yang ada pada jaringan SDN ini dengan

menggunakan ONOS. Hasil dari implementasi pengujian" ini dapat dilihat pada Table 1.

Tabel 1 merupakan hasil yang didapatkan dengan nilai rata rata *jitter* 0.1585 ms, *packet loss* 7.428 %, *throughput* UDP 2.3819 bit/sec, *throughput* UTP 11.912 bit/sec, dan *bandwith* rata-rata dari pengujian ini bernilai 9.96 Mbps.

Table 2 *Perfomance Device* Switch OpenFlow ONOS

Switch	*Packet Received*	*Package Sent*	*Byte Received*	*Byte Sent*
1	42,254	42,296	3,452,549	3,456,529
2	42,109	42,131	3,435,257	3,437,617
3	42,119	42,147	3,436,373	3,439,185
4	49,565	42,128	3,435,077	3,437,669
5	2,300,514	2,703,668	58,259,174,177	116,391,920,821
6	14,533	55,734	1,183,575	4,527,800
7	14,538	55,747	1,183,907	4,528,802
8	14,538	55,766	1,183,985	4,530,392
9	14,548	14,054,738	27,422	4,533,186
10	14,553	43,277	1,185,097	113,101,369,060
Total	2,549,271	17,137,632	58,277,697,419	229,525,181,061

Table 2 merupakan hasil analisi *perfomance* dari tiap switch openflow pada jaringan ini dimana jika di totalkan tiap hasil yang terdapat *Packet Received* = 2,549,271, *Packet Send*= 17,137,632, *Byte Received* =58,277,697,419, *Byte Sent* = 229,525,181,061.

5. Kesimpulan

Berdasarkan hasil dari penelitian ini yang berjudul "Implementasi dan analisis simulasi QOS pada jaringan SDN dan *perfomance* Device dengan menggunakan ONOS dan Iperf3" dapat disimpulkan bahwa. Dari 2 Pengujian yang diuji melalui *topology custom* yang dibuat dengan Bahasa python dengan API Mininet yang dimana pertama melakukan pengujian *Quality of* service pada mininet menggunakan *tools* iperf3 *telah* memperoleh *jitter*, *packet loss*, *throughput* UDP dan *thoughput* UTP serta *Bandwith* rata-rata dari tiap host yang ada pada topology ini yaitu 18 hosts dan pada pengujian kedua melakukan analisis terhadap device Openflow switch yang dimana menyaring seluruh packet yang melewati switch tersebut pada analisis ini telah memperoleh data seperti *Packet Received*, *Packet send*, *Byte Received* dan *Byte Sent*.

Daftar Pustaka

Deepak Nadig Anantha, Zhe Zhang, Byrav Ramamurthy, Brian Bockelman, Garhan Attebury and David Swanson (2016). SNAG: *SDN-managed Network Architecture* forGridFTP *Transfers*, SC16 – INDIS.

Bruno Astuto A. Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti (2014). *A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Network*, IEEE Communications Surveys & Tutorials, Volume 16, Nomor 3.

Rika Wulandari (2016). Analisis QoS (*Quality Of Service*) pada jaringan *internet* (Studi Kasus: UPT Loka Uji Teknik Penambangan Jampang Kulon – LIPI), Jurnal Teknik Informatika dan Sistem Informasi, Volume 2, Nomor 2.

Rohmat Tulloh (2017). Analisis Performansi VLAN Pada Jaringan Software Defined Network (SDN), Jurnal Infotel, Volume 9, Nomor 2.